Homework #7

See Canvas for HW #7 assignment due date. Complete all of the following problems. Ideally, the theoretical problems should be answered in a Markdown cell directly underneath the question. If you don't know LaTex/Markdown, you may submit separate handwritten solutions to the theoretical problems. Please do not turn in messy work. Computational problems should be completed in this notebook (using the R kernel is preferred). Computational questions may require code, plots, analysis, interpretation, etc. Working in small groups is allowed, but it is important that you make an effort to master the material and hand in your own work.

A. Theoretical Problems

A.1 (12 points) Let $Y_1, \ldots, Y_n \overset{i}{\sim} Poisson(\lambda_i)$. Show that, if $\eta_i = \beta_0$, then the maximum likelihood estimator of λ_i is $\widehat{\lambda}_i = \bar{Y}$, for all $i = 1, \ldots, n$.

First, note that if $\eta_i = \beta_0$ for all $i = 1, \ldots, n$, then we can drop the i subscript. That is, we are not estimating a different mean for each response because there are no predictors. From class, we know that the log-likelihood for Poisson regression is:

$$\ell(\beta_0) = \sum_{i=1}^{n} \left[y_i \eta - e^{\eta} - \log(y_i!) \right] = \sum_{i=1}^{n} \left[y_i \beta_0 - e^{\beta_0} - \log(y_i!) \right] = \sum_{i=1}^{n} \left[y_i \beta_0 - e^{\beta_0} - \log(y_i!) \right]$$

To find the MLE, we maximize the log-likelihood with respect to β_0; the maximizer is the MLE. To find the MLE of λ, we use the relationship that $\lambda = e^{\beta_0}$.

We take the derivative of ℓ, set it equal to zero, and solve for :

$$\frac{d\ell(\beta_0)}{d\beta_0} = \sum_{i=1}^{n} \left[y_i \beta_0 - e^{\beta_0} - \log(y_i!) \right] = \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} e^{\widehat{\beta}_0} \overset{set}{=} 0 \qquad \implies \sum_{i=1}^{n} y_i = \sum_{i=1}^{n} e^{\widehat{\beta}_0} \implies \sum_{i=1}^{n} y_i = n e^{\widehat{\beta}_0} \implies \frac{1}{n} \sum_{i=1}^{n} y_i = e^{\widehat{\beta}_0} \implies \bar{y} = e^{\widehat{\beta}_0} \qquad \implies \widehat{\beta}_0 =$$

So, the MLE (as a random variable) of β_0 is $\widehat{\beta}_0 = \log(\bar{Y})$. By the invariance property of MLEs, the MLE of $\lambda = e^{\beta_0}$ is

$$\widehat{\lambda} = e^{\widehat{\beta}_0} = e^{\log(\bar{Y})} = \bar{Y}.$$

B. Computational Problems

Problem B.1

The National Institute of Diabetes and Digestive and Kidney Diseases conducted a study of 768 adult female Pima Indians living near Phoenix, A.Z. The purpose of the study was the investigate factors related to diabetes.

Before we analyze these data, we should note that some have raised ethical issues with its collection and popularity in the statistics and data science community. We should think seriously about these concerns. For example, Maya Iskandarani wrote a brief piece on consent and privacy concerns raised by this dataset. After you familarize yourself with the data, we'll then turn to these ethical concerns.

(a) (8 points) Perform simple graphical and numerical summaries of the data. Can you find any obvious irregularities in the data? If so, take appropriate steps to correct these problems.

```
library(ggplot2)
library(dplyr)
pima = read.table("https://www.colorado.edu/amath/sites/default/files/attached-files/pima.txt",
                  sep = "\t", header = TRUE)

#Here's a description of the data: https://rdrr.io/cran/faraway/man/pima.html
```

```
# no missing values
sum(is.na(pima))

par(mfrow=c(3,3))
for (i in 1:9) hist(pima[,i], col = i, main = names(pima)[i])
# histograms show weirdness -- glucose, diastolic, triceps, BMI, and insulin should never be zero
par(mfrow=c(1,1))

# recode zeros to NAs for values that can't be zero
metricTraits = c('glucose', 'diastolic', 'triceps', 'bmi', 'insulin')
pima[metricTraits][pima[metricTraits]==0] = NA
pima = pima %>%
    mutate(test = as.factor(test))

summary(pima)
```

Some measurements are recorded as zero when clearly they shouldn't be (e.g., glucose). We should store these values as NA.

(b) (12 points) Fit a model with the result of the diabetes test as the response and all the other variables as predictors. Store this model as `glmod_pima`. Can you tell whether this model fits the data?

```
glmod_pima = glm(test ~ ., data = pima, family = binomial)
summary(glmod_pima)

par(mfrow = c(2,2)); plot(glmod_pima)
```

```
### BEGIN HIDDEN TESTS
library(testthat)
expect_is(glmod_pima,"lm")
glmod_pima_Solution = glm(test ~ ., data = pima, family = binomial)
expect_equal(glmod_pima$coefficients[1][[1]], glmod_pima_Solution$coefficients[1][[1]])
### END HIDDEN TESTS
```

In the case where the response is binary, $Y = 0, 1$, as opposed to $Y = 0, 1, \ldots, n$, residuals won't fill a normal distribution and deviance will not follow a chisquared distribution, so we won't have any test for model fit. You might split the data into a training and test set, and see how well the model does at predicting values in the test set.

(c) (14 points) Using the model above, write R code to calculate the difference in the odds of testing positive for diabetes for a woman with a BMI at the first quartile compared with a woman at the third quartile, assuming all other factors are held constant. Store your answer in a variable `x`.

Also, give a confidence interval for this difference, stored in a variable `ci`.

```
y = quantile(pima$bmi, c(.25,.75), na.rm=T)
y
x = exp(glmod_pima$coef['bmi']*(y[1]-y[2]))
x
c(x - qnorm(0.975)*0.02908 , x + qnorm(0.975)*0.02908)
se = summary(glmod_pima)$coefficients['bmi','Std. Error']
ci = exp((glmod_pima$coef['bmi']+c(1.96,-1.96)*se)*(27.5-36.6))
ci
```

```
### BEGIN HIDDEN TESTS
y_Solution = quantile(glmod_pima$data$bmi, c(.25,.75), na.rm=T)
x_Solution = exp(glmod_pima$coef['bmi']*(y_Solution[1]-y_Solution[2]))
expect_equal(x, x_Solution)
se_Solution = summary(glmod_pima)$coefficients['bmi','Std. Error']
ci_Solution = exp((glmod_pima$coef['bmi']+c(1.96,-1.96)*se_Solution)*(27.5-36.6))
expect_equal(ci, ci_Solution)
### END HIDDEN TESTS
```

Note above that, for bmi: 1st Qu.:28.40, 3rd Qu.:37.10. Let o_i be the odds of a woman getting diabetes at the i^{th} quantile.

$$\frac{o_1}{o_3} = \exp\left\{\log\left(o_1/o_3\right)\right\} = \exp\left\{\eta_1 - \eta_3\right\} = \exp\left\{\widehat{\beta}_{\text{bmi}}(27.5 - 36.6)\right\} \approx 0.526$$

So, adjusting for the other predictors, the odds of showing evidence of diabetes is 52.6% less for a woman in the 1st bmi quantile than in the 3rd bmi quantile. The CI is also calculated above.

(d) (STAT 5010 Students Only, 10 points) **Do women who test positive have higher diastolic blood pressures? Is the diastolic blood pressure significant in the regression model? Explain the distinction between the two questions and discuss why the answers are only apparently contradictory.**

```
#cor(glmod_pima$model)
lm_diastolic = lm(diastolic ~ test, data = pima)
summary(lm_diastolic)
```

Women who test positive do have a higher diastolic blood pressure, on average. However, the coefficient for diastolic is not significant in the model. One is a question about the result of the test conditional on diastolic pressure; the other is a question about diastolic blood pressure conditional on a test result. We know from Bayes' theorem that these are not the same!

(e) (5 points) Ethical Issues in Data Collection

Read Maya Iskandarani's piece on consent and privacy concerns raised by this dataset. Summarize those concerns here.

Iskandarani's concerns about this dataset are related to privacy and consent in the age of big data. The original pima study, from which the data came, was meant to last 10 years but ended up lasting 40, and years later, was archived by the University of California Irvine Machine Learning Repository. This archiving made the pima dataset a "standard" dataset for training statistical and machine learning algorithms on. Those who signed up for the study never could have known that they were going to be signing over their data to be used in these ways.

Problem B.2

The ships dataset gives the number of damage incidents and aggregate months of service for different types of ships broken down by year of construction and period of operation.

(a) (12 points) The code below splits the data into a training set (80% of the data) and a test set (the remaining 20%). Use the training set to develop an appropriate regression model for the rate of incidents, using type, period, and year as predictors (HINT: is this a count model or a rate model?). Store this model in `glmod_ships`.

```
library(MASS)
data(ships)
ships = ships[ships$service != 0,]
ships$year = as.factor(ships$year)
ships$period = as.factor(ships$period)


dim(ships)
set.seed(11)
n = floor(0.8 * nrow(ships))
index = sample(seq_len(nrow(ships)), size = n)

train = ships[index, ]
test = ships[-index, ]
head(train)
dim(train)
summary(train)
```

```
library(ggplot2)
#ggplot(ships, aes(year, incidents, col = type, size=service, shape = as.factor(period))) + geom_point()

glmod_ships = glm(incidents ~ type + period + year + offset(log(service)), family=poisson, data = train)
summary(glmod_ships)


#nbmod = glm.nb(incidents ~ ., data = ships)
#summary(nbmod)
```

```
### BEGIN HIDDEN TESTS
expect_is(glmod_ships,"glm")
glmod_ships_Solution = glm(incidents ~ type + period + year + offset(log(service)), family=poisson, data = train)
expect_equal(glmod_ships$coefficients[1][[1]], glmod_ships_Solution$coefficients[1][[1]])
### END HIDDEN TESTS
```

(b) (8 points) Use the model that you stored in glmod_ships to calculate the mean squared prediction error (MSPE) for the test set. Store the predicted MSPE in `mse_glmod_ships`.

Recall from earlier assignments that the MSE can give us a sense of how well the model does at predicting new observations. The predicted mean squared error (MSE) is defined as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2,$$

where y_i is the response in the test set, and $\hat{y}_i$ is the predicted response from glmod_ships, given the predictor values in the test set.

Note that the predict.glm() function can be helpful here. Just be sure to specify the type argument (HINT: do you want $\hat{y}_i$ to be on the scale of the linear predictor η, or the mean of the response?)

```
pred = predict.glm(glmod_ships, type = "response", test);
mse_glmod_ships = with(test, mean((incidents - pred)^2))

mse_glmod_ships
```

```
### BEGIN HIDDEN TESTS
pred_Solution = predict.glm(glmod_ships_Solution, type = "response", test);
mse_glmod_ships_Solution = with(test, mean((incidents - pred_Solution)^2))
#expect_equal(chisq_null, chisq_null_Solution)
expect_equal(mse_glmod_ships, mse_glmod_ships_Solution)
### END HIDDEN TESTS
```

(c) (9 points) Now construct a new regression model leaving out the year predictor. Store this model as `glmod_ships2`. Calculate the predicted MSPE for the test set using `glmod_ships2`. Decide which model is better - `glmod_ships` or `glmod_ships2` - and store your answer in `glmod_ships3`.

```
glmod_ships2 = glm(incidents ~ type + period + offset(log(service)), family=poisson, data = train)
summary(glmod_ships2)

pred2 = predict.glm(glmod_ships2, type = "response", test, na.rm = TRUE);
mse_glmod_ships2 = with(test, mean((incidents - pred2)^2))
mse_glmod_ships2

mse_glmod_ships2 - mse_glmod_ships
glmod_ships3 = glmod_ships
```

```
### BEGIN HIDDEN TESTS
glmod_ships2_solution = glm(incidents ~ type + period + offset(log(service)), family=poisson, data = train)
expect_equal(glmod_ships2, glmod_ships2_solution)
pred2_Solution = predict.glm(glmod_ships2, type = "response", test);
mse_glmod_ships2_Solution = with(test, mean((incidents - pred2)^2))
expect_equal(mse_glmod_ships2, mse_glmod_ships2_Solution)
expect_equal(glmod_ships3, glmod_ships)
### END HIDDEN TESTS
```

(d) (12 points) Let $\alpha = 0.05$. Conduct two χ^2 tests (using the deviance):

1. Test the adequacy of null model (store the p-value for this test in chisq_null); and
2. Test the adequacy of the glmod_ships model agaisnt the saturated model (store the p-value for this test in chisq_p).

What conclusions should you draw from these tests?

```
pr = ifelse(train$incidents*log(train$incidents/fitted(glmod_ships2, type = "response")) == "NaN",
            0, train$incidents*log(train$incidents/fitted(glmod_ships2, type = "response")));
with(train, 2*sum(pr - (incidents - fitted(glmod_ships2))))
chisq_null = 1-pchisq(124.731 , 26)
chisq_p = 1-pchisq(deviance(glmod_ships), 18)
chisq_null
chisq_p

### Chi-squared test
pcs = with(train, sum((incidents - fitted(glmod_ships, type = "response"))^2/fitted(glmod_ships, type = "response")));
pcs
1-pchisq(pcs,26)
```

```
### BEGIN HIDDEN TESTS
chisq_null_Solution = 1-pchisq(124.731, 26)
chisq_p_Solution = 1-pchisq(deviance(glmod_ships), 18)
expect_equal(chisq_null, chisq_null_Solution) #can we do expect_equal with some tolerance?
expect_equal(chisq_p, chisq_p_Solution)
### END HIDDEN TESTS
```

The p-value for the null model test is much smaller than α. Thus, we reject the null hypothesis that the null model is sufficient, and conclude that some predictors are necessary.

The p-value for the glmod_ships model test is larger than α). Thus, we fail to reject the null hypothesis H_0: the glmod_ships model is inadequate.

(e) (8 points) Plot the deviance residuals against the linear predictor η. Interpret this plot.

```
r = residuals(glmod_ships, type = "deviance")
f = predict(glmod_ships, type = "link")
df = data.frame(r,f)
ggplot(df) + geom_point(aes(x = f, y = r)) + theme_bw()
```

Under reasonable assumptions, the deviance residuals will be normally distributed around zero. This point does not provide strong evidence of nonnormality or structural deficiencies in our model.

(f) (STAT 5010 Students Only, 7 points) For some GLMs (including the type in this question!), *overdispersion* is sometimes a problem. *Overdispersion* occurs when the observed (data) variance is higher than expected, if the model is correct. Explore the two models above for evidence of overdispersion.

```
library(AER)
#this package has a function overdispersiontest(), which conducts an overdisperion test.
#If you use it, please clearly describe the test being used, including hypotheses, test statistic distribution,
#and conclusions
```

```
dispersiontest(glmod_ships)
dispersiontest(glmod_ships2)
d1 = deviance(glmod_ships2)/21
d2 = deviance(glmod_ships)/18
d1
d2
```

Neither model shows evidence of overdispersion.